A. Basic scheme functions: car, cdr, cons, append, lambda, if, cond, etc.
B. How Lets, Lambdas and Applications are evaluated by the system.
C. Map and Apply
D. Foldl and foldr
E. Unrestricted lambdas
F. Tail recursions, deep recursions and accumulator-passing style
G. Representing data structures
H. Trees
I. Environments and set!
J. Programming language issues
   a. State and set!
   b. Static vs dynamic binding
   c. Procedure calling mechanisms (call-by-value, name, reference, etc.)
K. Our interpreter project
   a. How specific expressions are parsed/evaluated in our Minischeme project
      including
      i. Let
      ii. Lambda
      iii. Procedure applications
   b. What would happen if we did something differently, such as changing from call-by-value to call-by-reference
O. Streams
P. Continuations and continuation-passing style
Q. call/cc
R. Nothing on Combinators